

## CLAIMS

What is claimed is:

1 1. A method of detecting a malware comprising the steps of:  
2       interrupting execution of a process that has been loaded for execution;  
3       scanning the process for a malware;  
4       allowing the process to execute, if no malware is found; and  
5       terminating execution of the process, if a malware is found.

1 2. The method of claim 1, wherein the process is associated with an  
2 application program.

1 3. The method of claim 1, wherein the process is loaded from at least one  
2 compressed, packed, or encrypted file.

1 4. The method of claim 1, wherein execution of the process comprises the  
2 step of:  
3       loading code for execution by the process from at least one compressed,  
4       packed, or encrypted file.

1 5. The method of claim 4, wherein the step of interrupting execution of the  
2 process comprises the step of:

- 3                   interrupting execution of the process when the process accesses at least
- 4    one file that is not needed to perform decryption, decompression, or unpacking.

1 6. The method of claim 5, wherein the at least one file that is not needed to  
2 perform decryption, decompression, or unpacking comprises a system library file.

1 7. The method of claim 5, wherein the at least one file that is not needed to  
2 perform decryption, decompression, or unpacking comprises an executable file  
3 not related to the process.

1 8. The method of claim 5, wherein the at least one file that is not needed to  
2 perform decryption, decompression, or unpacking comprises a data file not  
3 related to the process.

1 9. The method of claim 5, wherein the malware is a computer virus.

1 10. The method of claim 5, wherein the malware is a computer worm.

1 11. The method of claim 5, wherein the malware is a Trojan horse program.

1 12. The method of claim 5, further comprising the step of:

2 scanning the process for a malware before execution of the process.

1 13. A system for detecting a malware comprising:  
2 a processor operable to execute computer program instructions;  
3 a memory operable to store computer program instructions executable  
4 by the processor; and  
5 computer program instructions stored in the memory and executable to  
6 perform the steps of:  
7 interrupting execution of a process that has been loaded for execution;  
8 scanning the process for a malware;  
9 allowing the process to execute, if no malware is found; and  
10 terminating execution of the process, if a malware is found.

1 14. The system of claim 13, wherein the process is associated with an  
2 application program.

1 15. The system of claim 13, wherein the process is loaded from at least one  
2 compressed, packed, or encrypted file.

1 16. The system of claim 13, wherein execution of the process comprises the  
2 step of:

3                   loading code for execution by the process from at least one compressed,  
4                   packed, or encrypted file.

1   17.   The system of claim 16, wherein the step of interrupting execution of the  
2                   process comprises the step of:

3                   interrupting execution of the process when the process accesses at least  
4                   one file that is not needed to perform decryption, decompression, or unpacking.

1   18.   The system of claim 17, wherein the at least one file that is not needed to  
2                   perform decryption, decompression, or unpacking comprises a system library file.

1   19.   The system of claim 17, wherein the at least one file that is not needed to  
2                   perform decryption, decompression, or unpacking comprises an executable file  
3                   not related to the process.

1   20.   The system of claim 17, wherein the at least one file that is not needed to  
2                   perform decryption, decompression, or unpacking comprises a data file not  
3                   related to the process.

1   21.   The system of claim 17, wherein the malware is a computer virus.

1 22. The system of claim 17, wherein the malware is a computer worm.

1 23. The system of claim 17, wherein the malware is a Trojan horse program.

1 24. The system of claim 17, further comprising the step of:

2 scanning the process for a malware before execution of the process.

1 25. A computer program product for detecting a malware comprising:

2 a computer readable medium;

3 computer program instructions, recorded on the computer readable

4 medium, executable by a processor, for performing the steps of

5 interrupting execution of a process that has been loaded for execution;

6 scanning the process for a malware;

7 allowing the process to execute, if no malware is found; and

8 terminating execution of the process, if a malware is found.

1 26. The computer program product of claim 25, wherein the process is

2 associated with an application program.

1 27. The computer program product of claim 25, wherein the process is loaded

2 from at least one compressed, packed, or encrypted file.

1 28. The computer program product of claim 25, wherein execution of the  
2 process comprises the step of:

3 loading code for execution by the process from at least one compressed,  
4 packed, or encrypted file.

1 29. The computer program product of claim 28, wherein the step of  
2 interrupting execution of the process comprises the step of:

3 interrupting execution of the process when the process accesses at least  
4 one file that is not needed to perform decryption, decompression, or unpacking.

1 30. The computer program product of claim 29, wherein the at least one file  
2 that is not needed to perform decryption, decompression, or unpacking comprises  
3 a system library file.

1 31. The computer program product of claim 29, wherein the at least one file  
2 that is not needed to perform decryption, decompression, or unpacking comprises  
3 an executable file not related to the process.

1 32. The computer program product of claim 29, wherein the at least one file  
2 that is not needed to perform decryption, decompression, or unpacking comprises  
3 a data file not related to the process.

1 33. The computer program product of claim 29, wherein the malware is a  
2 computer virus.

1 34. The computer program product of claim 29, wherein the malware is a  
2 computer worm.

1 35. The computer program product of claim 29, wherein the malware is a  
2 Trojan horse program.

1 36. The computer program product of claim 29, further comprising the step  
2 of:  
3 scanning the process for a malware before execution of the process.